

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-160943

(43)Date of publication of application : 20.06.1997

(51)Int.Cl.

G06F 17/50

G01R 31/28

G06F 11/22

(21)Application number : 07-316219

(71)Applicant : PFU LTD

(22)Date of filing : 05.12.1995

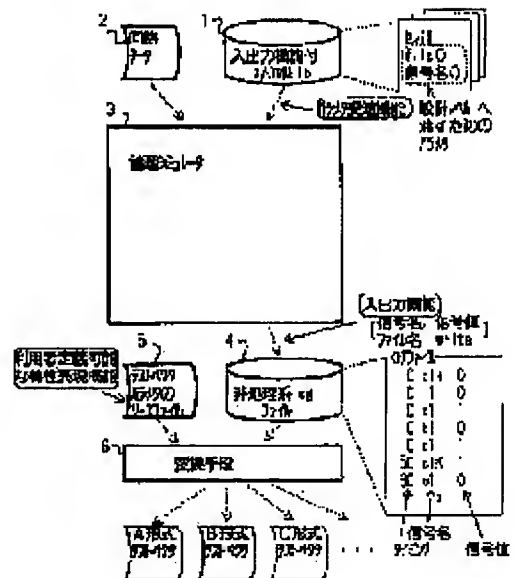
(72)Inventor : NIIDE HIROTAKE
NAKAMURA TAKAYO
YOSHIE HIROSHI

(54) METHOD FOR PREPARING TEST VECTOR IN LOGIC SIMULATOR AND SYSTEM THEREFOR

(57)Abstract:

PROBLEM TO BE SOLVED: To convert a logical verification result into each test vector form of each IC vender without being conscious of the difference of the IC vender and an AD tool.

SOLUTION: The parameter defining the signal name of the I/O cell of a design circuit is described in an I/O cell library 1 defining an input/output function. By using the parameter receiving delivering function that the circuit expression/hardware description language of a logic simulator 3, the parameter is received and delivered by the logic simulator 3 and the logic verification of circuit data 2 is performed. The I/O cell signal value of the design circuit which is obtained as the logic verification result and the signal name of each cell obtained based on the parameter are written in a non-processing system log file 4 by using the input/output function that the circuit expression/hardware description language has. Next, the test vectors of various kinds of forms are prepared by a conversion means 6 by utilizing the data stored in the source file 4 of a test vector parameter and the non-processing system log file 4.



* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1]A function for input/output procedure with a processor a simulator runs with an I/O function to be possible, and to deliver a parameter to a low rank from a design hierarchy's higher rank.

A characteristic table present function by user definition.

While being a test vector preparation method in a logic simulator having the above and creating circuit data of a designed circuit, A parameter which defines a signal name of the input / output cell of the above-mentioned designed circuit, and a bidirectional cell as an I/O cell library which defined an input output function is described, A parameter described by the above-mentioned I/O cell library is delivered to a logic simulator using a delivery function of the above-mentioned parameter, A logic simulator performs logic verification of the above-mentioned designed circuit using the above-mentioned circuit data and an I/O cell library, A signal value of the input / output / bidirectional cell of the above-mentioned designed circuit and a signal value of a control signal of a bidirectional cell which were acquired as a logic verification result, A signal name of each cell which might be based on the above-mentioned parameter is written in a non-processor log file using the above-mentioned I/O function, A test vector header is defined using a characteristic table present function by user definition of the above-mentioned circuit expression / hardware description language, and a test vector of various forms is created using data stored in this test vector header and the above-mentioned non-processor log file.

[Claim 2]A function for input/output procedure with a processor a simulator runs with an I/O function to be possible, and to deliver a parameter to a low rank from a design hierarchy's higher rank.

A characteristic table present function by user definition.

An I/O cell library which is a test vector preparing system in a logic simulator having the above, and described a parameter which defines a signal name of the input / output cell of a designed circuit, and a bidirectional cell, A non-processor log file which stores a logic verification result by a logic simulator, A test vector header definition file which defines a test vector header using a characteristic table present function by the above-mentioned user definition, It has a conversion method which creates a test vector of various forms from the above-mentioned non-processor log file and a test vector header definition file, A parameter described by the above-mentioned I/O cell library is delivered to a logic simulator using a delivery function of the above-mentioned parameter, A logic simulator performs logic verification of the above-mentioned designed circuit using circuit data and an I/O cell library of a designed circuit, A signal value of the input / output / bidirectional cell of the above-mentioned designed circuit and a signal value of a control signal of a bidirectional cell which were acquired as a logic verification result, Write a signal name of each cell which might be based on the above-mentioned parameter in the above-mentioned non-processor log file using the above-mentioned I/O function, and the above-mentioned test vector header definition file and a non-processor log file are used, A test vector of various forms is created by the above-mentioned conversion method.

[Translation done.]

* NOTICES *

JPO and INPIT are not responsible for any damages caused by the use of this translation.

- 1.This document has been translated by computer. So the translation may not reflect the original precisely.
- 2.**** shows the word which can not be translated.
- 3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention]When depending for IC manufacture on IC vendor, usually as a design data from IC vendor, It is requested that the logical data of IC which is the target of the above-mentioned manufacture, and the test data (test pattern) for testing this IC should be released (the test data which IC vendor demands is hereafter called test vector). When a commercial CAD tool (tool for performing logic verification, such as IC provided with the simulator, etc.) is used as a verification tool in the logic design verification at the time of designing IC, It cannot release to IC vendor by having to use the verification result as release data as it is, but must change into the test vector form that IC vendor demands a verification result.

[0002]Since IC vendor has an original test vector form, respectively, when a commercial CAD tool is used as mentioned above, the converting tool which changes the verification result into the test vector form of each IC vendor is needed, respectively. Since the I/O form of a CAD tool depends on each CAD tool, the above-mentioned converting tool must be prepared for every CAD tool.

[0003]This invention about the preparation method and system of a test vector in the logic simulator for changing the verification result obtained as mentioned above using the commercial CAD tool into the test vector which IC vendor demands, Especially this invention relates to the preparation method and system of a test vector which can change the verification result of a commercial CAD tool into the test vector form of each IC vendor, without an integrated circuit design person being conscious of the difference in IC vendor or a CAD tool.

[0004]

[Description of the Prior Art]Drawing 5 and drawing 6 are the figures showing the outline composition of the test vector form which IC vendor demands, and drawing 5 and drawing 6 have illustrated two different test vector forms. Drawing 5 is a shown figure the 1st test vector form (A form), and the test vector of A form, It comprises a test block etc. which described the test patterns to be a test vector header which shows a test data name, a logic circuit name, etc., and a timing specification part which specifies the positive/negative of the pulse given to each terminal of IC, pulse length, etc. A test block is a portion which describes the test pattern for testing IC which IC vendor manufactured.

In the example of the figure, each terminal name of IC was defined as the next of "FORM=", the logical level of the signal given to the terminal of IC by input description and output description is specified, and the signal value of the input output signal of each terminal described by the next of FORM=" is described by the portions of the above "input description" and "output description."

[0005]In the example of the figure, it is written as @12, CLK, BUSC, and — by the next of "FORM=", moreover — although written as "N", "0", "P", and — by the portion of input description / output description ("0", "1", "P", and "N") [and] "L" "H", "Z", and "X" this showing an output, Behind the space 12 (@ shows a space), to a CLK terminal, give "0" to "N" and a BUSC terminal and a test pattern is given like "P" and — to terminal CLR, According to it,

it means that an output like "L", "L", -- "H" occurs (the test pattern sequence at the time of being about a transverse direction). [from the corresponding terminal specified by FORM="] [show and] Party eyes are a test pattern sequence at the time of the test pattern sequence at the time of the test pattern sequence at the 1st time and the 2nd line being the 2nd and the 3rd line being the 3rd, and --. The portion which there are an input terminal, an output terminal, and a bidirectional terminal in the terminal of IC, and is shown as "both directions" with the figure here is a signal value of the bidirectional terminal to which input and output are switched by a control signal.

[0006]Drawing 6 is a figure showing the 2nd test vector form (B form), and the test vector of B form comprises a test block etc. which described the test patterns to be a test vector header which shows a test data name, a logic circuit name, etc., a terminal definition part, and a timing specification part. a terminal is defined by the terminal definition part in B form -- for example, each input terminal -- "CLK1 REF=1", "SMC1 REF=2", --, ** -- it defining like (in CKL1 terminal, "1" and SMC1 terminal means "2" and corresponding, without --), and an input terminal, ["IN" and] As for the output terminal, "OUT" and a bidirectional terminal are shown as "IO." Have described the signal value of each terminal defined as mentioned above by the test block, and in the example of the figure. Timing TM1 shows "0" and the case where -- is given to the CLK terminal (REF=1) at P (positive pulse) and SMC1 terminal (REF=2) (the definition of "P", "0", --, is the same as drawing 5).

[0007]As shown in drawing 5 and drawing 6, test vector form differs in form according to IC vendor, In releasing to IC vendor as it is by using as release data the verification result obtained using the CAD tool (tool for performing logic verification, such as IC provided with the simulator, etc.) at the time of an integrated circuit design, It is necessary to change the above-mentioned verification result into the test vector form corresponding to each IC vendor.

[0008]Drawing 7 is a figure showing the composition of the conventional conversion system for changing into the test vector form of each IC vendor the logic verification result obtained with the CAD tool. The circuit data file which stored the circuit data of IC in which 51 is the target of logic verification in the figure, 52 and 53 are the 1st and 2nd processors (the above mentioned CAD tool) containing the simulator for verifying the logic of the above-mentioned IC, etc., The 1st and 2nd processor 52 and 53 is carrying out normal use of what is marketed, for example, the CAD tool by X company in the processor X and the processor Y are the CAD tools by Y company. The compilers 52a and 53a which the processors 52 and 53 compile the circuit data stored in the above-mentioned circuit data file 51, and generate the configuration data 52b and 53b of the executable code corresponding to circuitry, It has the simulators 52c and 53c which carry out the simulation of the logic operation of IC based on the above-mentioned configuration data 52b and 53b, and obtain a logic verification result, and the simulator log files 52d and 53d which store the above-mentioned simulation result. 55-60 are conversion methods which change into the test vector of each company form the logic verification result stored in the above-mentioned simulator log files 52d and 53d, and these conversion methods 55-60 are prepared every processor 52 and processor 53 again for every A-C which is IC vendor.

[0009]In the figure, an integrated circuit design person uses the processor 52 by X company, In performing logic verification of IC, storing the logic verification result in the simulator log file 52d and requesting manufacture of this IC from A company, A firm business circuit tester information and the above-mentioned logic verification result are used, and it creates the test vector of A company form by the conversion method 55. Similarly, an integrated circuit design person uses the processor 53 by Y company, and performs logic verification of IC, In storing the logic verification result in the simulator log file 53d and requesting manufacture of this IC from A company, A firm business circuit tester information and the above-mentioned logic verification result are used, and it creates the test vector of A company form by the conversion method 58. That is, the integrated circuit design person has to select a suitable conversion method according to the processor and IC vendor which were used, and has to create the test vector of each company form.

[0010]

[Problem(s) to be Solved by the Invention]As mentioned above, in the former, the conversion

method had to be prepared for every processor and IC vendor, and the integrated circuit design person had to create the test vector using the suitable conversion method according to the processor and IC vendor which were used. Usually, the form of the log file of a commercial CAD tool is as the above-mentioned conversion method, when using two or more CAD tools and requesting manufacture of IC from two or more IC vendors, since it differs for every CAD tool, [The number of simulator log file forms] x A number equivalent to [the number of IC vendor forms] of conversion methods were needed, and the designer needed to choose the suitable conversion method according to the CAD tool and IC vendor which were used out of those conversion methods. For this reason, the above-mentioned converting operation had become a big burden for a designer.

[0011] This invention is made in consideration of the problem of the above-mentioned conventional technology, and the purpose of this invention, In one procedure defined beforehand, without an integrated circuit design person being conscious of the difference in IC vendor or a CAD tool. By creating the log file which stores a verification result, it is providing the preparation method and system of a test vector in the logic simulator who can change the result of the logic verification by a logic simulator into the test vector form of each IC vendor.

[0012]

[Means for Solving the Problem] Drawing 1 is an outline lineblock diagram of this invention. In the figure, 1 is an I/O cell library which defines a function of an I/O cell of a designed circuit, and a parameter which defines a signal name of an I/O cell as the I/O cell library 1 is described. 2 is the file which stored circuit data of a designed circuit. 3 is a logic simulator and the logic simulator 3, A function for input/output procedure with a processor a simulator runs with an I/O function to be possible, and to deliver a parameter to a low rank from a design hierarchy's higher rank, The simulation of the circuit model expressed / described with circuit expression / hardware description language provided with a characteristic table present function by user definition is carried out, and a logic verification result is outputted. 4 is a non-processor log file and a signal value and a signal name of an I/O cell of a designed circuit are written in the non-processor log file 4 among the logic simulator's 3 logic verification results using the above-mentioned input output function. 5 is a source file of a test vector parameter created using a characteristic table present function by the above-mentioned user definition — header information of a test vector — **** — last ** 6 is a conversion method and creates a test vector of various forms based on the source file 5 of data stored in the above-mentioned non-processor log file 4, and a test vector parameter.

[0013] In order to solve said technical problem, an invention of claims 1 and 2 of this invention, As shown in drawing 1, to the I/O cell library 1 which defined an input output function. A parameter which defines a signal name of the input / output cell of a designed circuit, and a bidirectional cell is described, Deliver this parameter to the logic simulator 3 using a PARATAME delivery function which the logic simulator's 3 circuit expression / hardware description language have, and logic verification is performed, A signal value of the input / output / bidirectional cell of the above-mentioned designed circuit and a signal value of a control signal of a bidirectional cell which were acquired as a logic verification result, A signal name of each cell which might be based on the above-mentioned parameter is written in the non-processor log file 4 using an input output function, A test vector of various forms is created by the conversion method 6 using data stored in a test vector header and the above-mentioned non-processor log file which are defined by the source file 4 of the above-mentioned test vector parameter.

[0014] A parameter delivery function which circuit expression / hardware description language has as mentioned above in this invention, Data which generated a non-processor log file which traced a signal of a portion of an I/O cell of designed circuit data, and stored a signal value of an I/O cell of a designed circuit, a signal name, etc. using an input output function, and was stored in a non-processor, Since a test vector of various forms was created based on specified IC vendor information, the circuit designer can create a test vector of various forms, without being conscious of a logic simulator and each vendor form to be used.

[0015]

[Embodiment of the Invention] Drawing 2 is a figure showing the composition of the system of the

example of this invention. In the figure, 51 is the circuit data file which stored the circuit data of IC which is the target of the logic verification shown in said drawing 3, and the source program which shows the circuitry of a designed circuit is stored in the circuit data file 51. 52 and 53 are the 1st and 2nd processors containing the simulator for verifying the logic of the above-mentioned IC, etc. The compilers 52a and 53a which the processors 52 and 53 compile circuit data, the I/O cell library mentioned later, etc., and generate the configuration data 52b and 53b, It has the simulators 52c and 53c which carry out the simulation of the logic operation of IC based on the above-mentioned configuration data 52b and 53b, and obtain a logic verification result, and the simulator log files 52d and 53d which store the above-mentioned simulation result.

[0016]Here, the above-mentioned processors 52 and 53 in this invention are premised on the thing provided with the following feature. That is, circuit expression (or hardware language for describing circuit expression) satisfies the following requirements, and is provided with the function which carries out digital simulation of the circuit model using the circuit expression.

(1) The input/output procedure of an I/O function and the processor the simulator runs be possible. Namely, a thing provided with the function to read a signal name, a signal value, etc. from a predetermined file, or to write them to a predetermined file (this function is hereafter called input output function). This function is used in order to write a signal name, a signal value, etc. in a non-processor log file (a test vector is created using the log information stored in this file) so that it may mention later.

[0017](2) There needs to be a delivery mechanism of the parameter from a design hierarchy's higher rank to a low rank. Namely, the thing for which parameters, such as a signal name which the design hierarchy's higher rank defined, can be delivered to a design hierarchy's low rank (this function is hereafter called parameter delivery function). The parameters (signal name etc.) described by the source program of the I/O cell library can be made to reflect in a logic simulation result log with this function, so that it may mention later.

[0018](3) A user can define the characteristic as circuit expression. That is, the declaration which can write the contents which a test vector header should define is prepared, and using this function, the test vector header of specification IC vendor form is created so that it may mention later (this function is hereafter called test vector parameter description function). In the simulator for verifying the logic of a digital circuit, Usually, VHDL is used as a hardware description language and as the above (1), "REDA" of VHDL, "WRITE", etc. can be used and it is the above (2). If it carries out, The delivery function of a signal name with the generic parameter in VHDL can be used, and it is the above (3) further. If it carries out, "ATTRIBUTE" (attribute definition command) of VHDL can be used. In the following explanation, the ***** case for VHDL is explained as a hardware description language.

[0019]10 and 20 are portions added in this invention, and the input file which 10 becomes from the test vector parameter definition file 11 and the I/O cell library 12 with an input output function, and 20 are outputting parts which consist of the non-processor log file 21 and the conversion method 22. The correspondence relation of the terminal name and test pattern in a test vector, etc. are defined by the parameter definition file 11 of the input file part 10. The I/O cell library 12 with an input output function, The result of being the file which described the function of the I/O cell (it mentions later) of IC in the VHDL language, and having described the parameter here, and having compiled this I/O cell library 12 so that it might mention later, A logic simulation is performed using the result of having compiled the source program (design data) of circuit data, and a signal name etc. are made to reflect in the non-processor log file 21.

[0020]Among the logic verification results by the processors 52 and 53, the signal value of the I/O cell of IC gives the signal name to the non-processor log file 21 of the outputting part 20, and is stored in it, and the test vector of each company form is created based on this. The signal value of the above-mentioned I/O cell is a signal value of each pin of IC, and the signal value of the control signal which controls a bidirectional I/O cell is also stored about a bidirectional terminal. By 22 being a conversion method which changes into the test vector of each company form the signal name of the I/O cell stored in the above-mentioned non-processor log file 21, logic, etc., and giving the parameter 31 which generates a test vector header part to the

conversion method 22, The conversion method 22 generates the test vector according to IC vendor form. Description of the above-mentioned parameter 31 uses the test vector parameter description function of the above mentioned VHDL language. Thereby, the integrated circuit design person can describe the parameter corresponding to a test vector header part, without being conscious of the test vector form of a CAD tool or each company.

[0021]Although it is possible to lower all the signal values (not only an input output signal but each signal value of the inner device of IC is included) of a simulation result to the simulator log files 52d and 53d provided in each processors 52 and 53, Since data required for test vector creation is a signal value of the output pin of IC, and all simulation results are not required reasons and the forms of the above-mentioned simulator log file differ for every CAD tool, In order to create the test vector of each company form from a simulator log file, as shown in said drawing 7, the conversion method for every CAD tool is needed.

[0022]On the other hand, the log file 22 independent of the form of a simulator of having stored data required for test vector creation is generable by forming the non-processor log file 21 and storing the signal name of the I/O cell of IC, a signal value, etc. here like this invention. And the test vector of each company form can be created only by giving the above-mentioned parameter 31 to it, if the function to change the above-mentioned log data into the test vector of each company form according to specification of the parameter 31 is given to the conversion method 22. And if it enables it to perform systematically description of the I/O cell library 12 with an input output function, description of circuit data, description of the above-mentioned parameter 31, etc. in a VHDL language, an integrated circuit design person, The test vector of each company form can be created without being conscious of the difference among the processors 52 and 53, the difference in each company vector form, etc.

[0023]Drawing 3 is a figure showing an example of the I/O cell model of LSI. The logic function circuit where LSI usually performs a desired logical operation etc. as shown in drawing 3, It has the I/O cell containing the output cell for outputting the output of the input cell which gives the input signal given from an external pin to the above-mentioned logic function circuit, and a logic function circuit to an external pin, the bidirectional cell by which a signal direction is controlled by a control signal, etc. Data required for test vector creation is the signal name (for example, i1 in the figure, o1) and signal value of the above-mentioned I/O cell. In the case of a bidirectional cell, a signal direction (are they an input state or an output state?) is needed with the signal value. Since the signal direction of a bidirectional cell becomes settled with the signal value of the control signal (c1 of the figure) given to a bidirectional cell, data required about a bidirectional cell is the signal name (for example, b1, c1) and signal value of each signal. You make it reflected in the output of the processors 52 and 53 in this example, by the parameter delivery function of VHDL which described above the parameter described by the I/O cell library 12 with an input output function, as shown in drawing 3, A signal name is written in the non-processor log file 21 with the logic verification result of the processors 52 and 53 using the above mentioned input output function of VHDL.

[0024]Drawing 4 is a figure explaining operation of the system of the example shown in drawing 2, and explains the conversion process procedure in this example by drawing 3 and drawing 4. First, the parameter for passing a design level is described to the I/O cell library (source program) 31. Next, a circuit design is performed using this I/O cell library 31, and the configuration file (source of a designed circuit) 32 is created (Step S1). As shown in the figure by this, in a parameter The actual signal name i1 (signal name of an input cell) of each I/O cell, The configuration file (source program) 32 containing o1 (signal name of an output cell), b1 (signal name of a bidirectional cell), c1 (signal name of the control signal of a bidirectional cell), etc. is created. Next, the above-mentioned configuration file 32 and the I/O cell library 31 are compiled by the existing processor, and the object program 33 of an I/O cell library and the object program 34 of a designed circuit are obtained (Step S2). And the signal value which performed the logic simulation by the existing processor using the above-mentioned object programs 33 and 34 (Step S3), and was acquired by logic verification using the above mentioned input output function is written in the non-processor log file 36 with a signal name.

[0025]The data written in the non-processor log file 36 is a signal value at each time of each I/O

cell (MYURESHON result), and a signal name, as shown in the figure. In the example of the figure, the control signal c1 which shows the signal value of the clock terminal CLK at each time (0, 50, 80, —), the signal value of the output cell o1, the signal value of the input cell i1, the signal value of the bidirectional cell b1, and the signal direction of a bidirectional cell is shown. With the figure, the case where the control signal c1 of a bidirectional cell changes from high level to a low level on the way is illustrated, and the signal direction of the bidirectional cell b1 is changing at this time. While creating the non-processor log file 36 as mentioned above, the test vector source file 35 which described test vector head sections, such as a designer and a circuit name, using the test parameter description function of a VHDL language is created. And the test vector of each vendor form as shown in drawing 5 and drawing 6 is created from the above-mentioned non-processor log file 36 and the test vector source file 35 (step S4).

[0026]

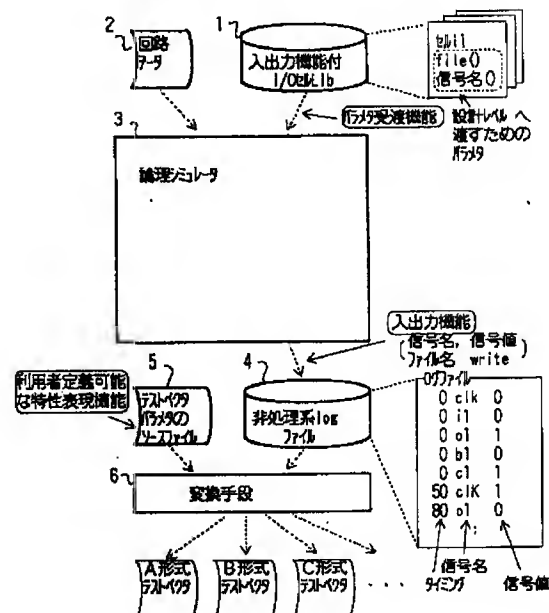
[Effect of the Invention]The parameter delivery function which circuit expression / hardware description language has in this invention as explained above, The data which generated the non-processor log file which stored the signal value of the I/O cell of a designed circuit, the signal name, etc. using the input output function, and was stored in the non-processor, Since the test vector of various forms was created based on specified IC vendor information, the circuit designer can create the test vector of various forms, without being conscious of the logic simulator and each vendor form to be used. For this reason, two or more conversion methods which embraced each vendor form like before are established, Since it becomes unnecessary to choose the logic simulator who uses it, and the conversion method needed out of the conversion method of this plurality according to each vendor form and various kinds of test vectors can be created in one expression and a procedure, reduction of an integrated circuit design person's burden can be carried out substantially.

[Translation done.]

(11)特許出願公開番号

(43)公開日 平成9年(1997)6月20日

審査請求 未請求 請求項の数2 OL (全 9 頁)



【特許請求の範囲】

【請求項 1】 入出力関数を持ちシミュレータが走行する処理系との入出力手続きが可能であって、設計階層の上位から下位へパラメタを受け渡す機能と、利用者定義による特性表現機能とを備えた回路表現／ハードウェア記述言語で表現／記述された回路モデルをシミュレーションする論理シミュレータの論理検証結果を利用して、各種形式のテストベクタを作成するテストベクタ作成方法であって、

設計回路の回路データを作成するとともに、入出力機能を定義した I/O セルライブラリに、上記設計回路の入力／出力セルおよび双方向セルの信号名を定義するパラメタを記述し、

上記 I/O セルライブラリに記述されたパラメタを上記パラメタの受け渡し機能を用いて論理シミュレータに受け渡し、上記回路データおよび I/O セルライブラリを用いて論理シミュレータにより上記設計回路の論理検証を行い、

論理検証結果として得られた上記設計回路の入力／出力／双方向セルの信号値および双方向セルの制御信号の信号値と、上記パラメタに基づき得られた各セルの信号名を、上記入出力関数を用いて非処理系ログファイルに書き込み、

上記回路表現／ハードウェア記述言語の利用者定義による特性表現機能を用いて、テストベクタヘッダを定義し、該テストベクタヘッダと上記非処理系ログファイルに格納されたデータを利用して、各種形式のテストベクタを作成することを特徴とする論理シミュレータにおけるテストベクタ作成方法。

【請求項 2】 入出力関数を持ちシミュレータが走行する処理系との入出力手続きが可能であって、設計階層の上位から下位へパラメタを受け渡す機能と、利用者定義による特性表現機能とを備えた回路表現／ハードウェア記述言語で表現／記述された回路モデルをシミュレーションする論理シミュレータの論理検証結果を利用して、各種形式のテストベクタを作成するテストベクタ作成システムであって、

設計回路の入力／出力セルおよび双方向セルの信号名を定義するパラメタを記述した I/O セルライブラリと、論理シミュレータによる論理検証結果を格納する非処理系ログファイルと、上記利用者定義による特性表現機能を用いてテストベクタヘッダを定義するテストベクタヘッダ定義ファイルと、上記非処理系ログファイルとテストベクタヘッダ定義ファイルから各種形式のテストベクタを作成する変換手段とを備え、

上記 I/O セルライブラリに記述されたパラメタを上記パラメタの受け渡し機能を用いて論理シミュレータに受け渡し、設計回路の回路データおよび I/O セルライブラリを用いて論理シミュレータにより上記設計回路の論

論理検証結果として得られた上記設計回路の入力／出力／双方向セルの信号値および双方向セルの制御信号の信号値と、上記パラメタに基づき得られた各セルの信号名を、上記入出力関数を用いて上記非処理系ログファイルに書き込み、上記テストベクタヘッダ定義ファイルと、非処理系ログファイルを用いて、上記変換手段により各種形式のテストベクタを作成することを特徴とする論理シミュレータにおけるテストベクタ作成システム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 IC 製造を IC ベンダに依存する場合、通常、IC ベンダから設計データとして、上記製造の対象となる IC の論理データと、該 IC をテストするためのテストデータ（テストパターン）をリリースするように要請される（IC ベンダが要請するテストデータを、以下、テストベクタという）。IC を設計する際の論理設計検証において、検証ツールとして市販の CAD ツール（シミュレータを備えた IC 等の論理検証等を行うためのツール）を使用した場合、その検証結果をそのままリリースデータとして IC ベンダにリリースすることはできず、検証結果を IC ベンダが要請するテストベクタ形式に変換しなければならない。

【0002】 IC ベンダはそれぞれ独自のテストベクタ形式を持つため、上記のように市販の CAD ツールを使用した場合には、その検証結果を各 IC ベンダのテストベクタ形式に変換する変換ツールがそれぞれ必要となる。また、CAD ツールの入出力形式は、それぞれの CAD ツールに依存するため、上記変換ツールは CAD ツール毎に用意しなければならない。

【0003】本発明は上記のように市販の CAD ツールを使用して得た検証結果を IC ベンダが要請するテストベクタに変換するための論理シミュレータにおけるテストベクタの作成方法およびシステムに関し、特に本発明は、IC 設計者が IC ベンダや CAD ツールの違いを意識することなく、市販の CAD ツールの検証結果を各 IC ベンダのテストベクタ形式に変換することができるテストベクタの作成方法およびシステムに関するものである。

【0004】

【従来の技術】図 5、図 6 は IC ベンダが要請するテストベクタ形式の概略構成を示す図であり、図 5、図 6 は 2 つの異なったテストベクタ形式を例示している。図 5 は第 1 のテストベクタ形式（A 形式）を示す図であり、A 形式のテストベクタは、テストデータ名、論理回路名等を示すテストベクタヘッダと、IC の各端子に与えるパルスの正負、パルス長等を指定するタイミング指定部と、テストパターンを記述したテストブロック等から構成される。テストブロックは、IC ベンダが製造した IC をテストするためのテストパターンを記述する部分で

子名を定義し、入力記述、出力記述で IC の端子へ与える信号の論理レベルを指定しており、FORM=" の次に記述される各端子の入出力信号の信号値が上記「入力記述」「出力記述」の部分に記述されている。

【0005】同図の例では、" FORM=" の次に @ 1 2, CLK, BUSC, ... と表記され、また、入力記述／出力記述の部分に " N", " 0", " P", ..., と表記されているが (" 0", " 1", " P", " N" が入力, " L", " H", " Z", " X" が出力を表す)、これは、スペース 1 2 (@ はスペースを示す) の後に、CLK 端子へ " N", BUSC 端子へ " 0", CLR 端子へ " P", ... のようにテストパターンを与え、それに
10 応じて、" FORM=" で指定されている対応した端子から " L", " L", ..., " H" のような出力が発生することを意味している(横方向がある時点におけるテストパターン列を示し、一行目は第 1 の時点におけるテストパターン列、2 行目が第 2 の時点におけるテストパターン列、3 行目が第 3 の時点におけるテストパターン列、... である)。ここで、IC の端子には入力端子と出力端子と双方向端子があり、同図で「双方向」として示
20 されている部分は制御信号により入出力が切り換えられる双方向端子の信号値である。

【0006】図 6 は第 2 のテストベクタ形式 (B 形式) を示す図であり、B 形式のテストベクタは、テストデータ名、論理回路名等を示すテストベクタヘッダと、端子定義部と、タイミング指定部と、テストパターンを記述したテストブロック等から構成される。B 形式において、端子は端子定義部で定義され、例えば、各入力端子は " CLK1 REF=1", " SMC1 REF=2", ..., のように定義され (CKL1 端子が " 1", SMC1 端子が " 2", ..., に対応することを意味している)、入力端子は " IN", 出力端子は " OUT", 双方向端子は、" IO" として示されている。テストブロックでは、上記のように定義された各端子の信号値を記述しており、同図の例では、タイミング TM1 で CLK 端子 (REF=1) に P (正パルス)、SMC1 端子 (REF=2) に " 0", ..., を与える場合を示している (" P", " 0", ..., の定義は図 5 と同じ)。

【0007】図 5, 図 6 に示すようにテストベクタ形式は IC ベンダに応じて形式が異なっており、IC 設計時に CAD ツール (シミュレータを備えた IC 等の論理検証等を行うためのツール) を使用して得た検証結果をそのままリリースデータとして IC ベンダにリリースする場合には、上記検証結果を各 IC ベンダに対応したテストベクタ形式に変換することが必要となる。

【0008】図 7 は CAD ツールにより得た論理検証結果を各 IC ベンダのテストベクタ形式に変換するための従来の変換システムの構成を示す図である。同図において、51 は論理検証の対象となる IC の回路データを格

論理を検証するためのシミュレータ等を含む第 1 および第 2 の処理系 (前記した CAD ツール) であり、第 1、第 2 の処理系 52, 53 は市販されているものを通常使用しており、例えば、処理系 X は X 社製の CAD ツール、処理系 Y は Y 社製の CAD ツールである。処理系 52, 53 は、上記回路データ・ファイル 51 に格納された回路データをコンパイルして回路構成に対応した実行形式のコンフィグレーションデータ 52b, 53b を生成するコンパイラ 52a, 53a と、上記コンフィグレーションデータ 52b, 53b に基づき IC の論理動作をシミュレーションして論理検証結果を得るシミュレータ 52c, 53c と、上記シミュレーション結果を格納するシミュレータログファイル 52d, 53d を備えている。55~60 は上記シミュレータログファイル 52d, 53d に格納された論理検証結果を各社形式のテストベクタに変換する変換手段であり、該変換手段 55~60 は処理系 52, 処理系 53 毎に、また、IC ベンダである A~C 社毎に用意されている。

【0009】同図において、例えば、IC 設計者が X 社製の処理系 52 を使用して、IC の論理検証を行い、シミュレータログファイル 52d にその論理検証結果を格納し、該 IC の製造を A 社に依頼する場合には、A 社用テスト情報と上記論理検証結果を使用して、変換手段 55 により A 社形式のテストベクタを作成する。また、同様に、IC 設計者が Y 社製の処理系 53 を使用して、IC の論理検証を行い、シミュレータログファイル 53d にその論理検証結果を格納し、該 IC の製造を A 社に依頼する場合には、A 社用テスト情報と上記論理検証結果を使用して、変換手段 58 により A 社形式のテストベクタを作成する。すなわち、IC 設計者は使用した処理系と IC ベンダに応じて適切な変換手段を選定し、各社形式のテストベクタを作成しなければならない。

【0010】

【発明が解決しようとする課題】以上のように従来においては、各処理系、IC ベンダ毎に変換手段を用意し、IC 設計者は、使用した処理系と IC ベンダに応じた適切な変換手段を用いてテストベクタを作成しなかった。通常、市販の CAD ツールのログファイルの形式は CAD ツール毎に異なっているから、複数の CAD ツールを使用して、複数の IC ベンダに IC の製造を依頼する場合には、上記変換手段として、〔シミュレータログファイル形式の数〕×〔IC ベンダ形式の数〕に相当する数の変換手段が必要となり、設計者はそれらの変換手段の中から、使用した CAD ツール、IC ベンダに応じた適切な変換手段を選択する必要があった。このため、上記変換作業は、設計者にとって大きな負担となっていた。

【0011】本発明は上記した従来技術の問題点を考慮してなされたものであって、本発明の目的は、IC 設計

く、予め定めた一つの手順で、検証結果を格納するログファイルを作成することにより、論理シミュレータによる論理検証の結果を各 IC ベンダのテストベクタ形式に変換することができる論理シミュレータにおけるテストベクタの作成方法およびシステムを提供することである。

【0012】

【課題を解決するための手段】図 1 は本発明の概略構成図である。同図において、1 は設計回路の I/O セルの機能を定義する I/O セルライブラリであり、I/O セルライブラリ 1 に I/O セルの信号名を定義するパラメタが記述される。2 は設計回路の回路データを格納したファイルである。3 は論理シミュレータであり、論理シミュレータ 3 は、入出力関数を持ちシミュレータが走行する処理系との入出力手続きが可能であって、設計階層の上位から下位へパラメタを受け渡し機能と、利用者定義による特性表現機能とを備えた回路表現/ハードウェア記述言語で表現/記述された回路モデルをシミュレーションし、論理検証結果を出力する。4 は非処理系ログファイルであり、非処理系ログファイル 4 には、上記入出力機能を使用して論理シミュレータ 3 の論理検証結果の内、設計回路の I/O セルの信号値とその信号名が書き込まれる。5 は上記利用者定義による特性表現機能を利用して作成したテストベクタパラメタのソースファイルでありテストベクタのヘッダ情報が格納される。6 は変換手段であり、上記非処理系ログファイル 4 に格納されたデータとテストベクタパラメタのソースファイル 5 に基づき各種形式のテストベクタを作成する。

【0013】前記課題を解決するため、本発明の請求項 1、2 の発明は、図 1 に示すように、入出力機能を定義した I/O セルライブラリ 1 に、設計回路の入力/出力セルおよび双方向セルの信号名を定義するパラメタを記述し、該パラメタを論理シミュレータ 3 の回路表現/ハードウェア記述言語が持つパラメタ受け渡し機能を用いて論理シミュレータ 3 に受け渡して論理検証を行い、論理検証結果として得られた上記設計回路の入力/出力/双方向セルの信号値および双方向セルの制御信号の信号値と、上記パラメタに基づき得られた各セルの信号名を、入出力機能を用いて非処理系ログファイル 4 に書き込み、上記テストベクタパラメタのソースファイル 4 で定義されるテストベクタヘッダと上記非処理系ログファイルに格納されたデータを利用して、変換手段 6 により各種形式のテストベクタを作成するようにしたものである。

【0014】本発明においては、上記のように、回路表現/ハードウェア記述言語が持つパラメタ受渡機能、入出力機能を用いて、設計回路データの I/O セルの部分の信号をトレースして設計回路の I/O セルの信号値、信号名等を格納した非処理系ログファイルを生成し、非

報に基づき各種形式のテストベクタを作成するようにしたので、回路設計者は、使用する論理シミュレータ、各ベンダ形式を意識することなく、各種形式のテストベクタを作成することができる。

【0015】

【発明の実施の形態】図 2 は本発明の実施例のシステムの構成を示す図である。同図において、51 は前記図 3 に示した論理検証の対象となる IC の回路データを格納した回路データ・ファイルであり、回路データ・ファイル 51 には、設計回路の回路構成を示すソースプログラムが格納される。52、53 は上記 IC の論理を検証するためのシミュレータ等を含む第 1 および第 2 の処理系である。処理系 52、53 は、回路データ、後述する I/O セルライブラリ等をコンパイルしてコンフィグレーションデータ 52b、53b を生成するコンパイラ 52a、53a と、上記コンフィグレーションデータ 52b、53b に基づき IC の論理動作をシミュレーションして論理検証結果を得るシミュレータ 52c、53c と、上記シミュレーション結果を格納するシミュレータログファイル 52d、53d を備えている。

【0016】ここで、本発明における上記処理系 52、53 は次の特徴を備えたものを前提としている。すなわち、回路表現（または、回路表現を記述するためのハードウェア言語）が次の要件を満たし、その回路表現を用いた回路モデルをデジタル・シミュレーションする機能を備えたものである。

(1) 入出力関数とそのシミュレータが走行する処理系との入出力手続きが可能であること。すなわち、信号名、信号値等を所定のファイルから read したり所定のファイルに write する機能を備えていること（以下、この機能を入出力機能という）。この機能は後述するように、非処理系ログファイル（このファイルに格納されたログ情報を利用してテストベクタを作成する）に信号名、信号値等を書き込むために利用される。

【0017】(2) 設計階層の上位から下位へのパラメタの受渡機構があること。すなわち、設計階層の上位で定義した信号名等のパラメタを設計階層の下位に受け渡すことができること（以下、この機能をパラメタ受渡機能という）。この機能により、後述するように、I/O セルライブラリのソースプログラムに記述されたパラメタ（信号名等）を論理シミュレーション結果ログに反映させることができる。

【0018】(3) 回路表現として利用者が特性を定義できること。すなわち、テストベクタヘッダで定義すべき内容を表記できるような宣言を用意し、この機能を利用して、後述するように、指定 IC ベンダ形式のテストベクタヘッダを作成する（以下、この機能をテストベクタパラメタ記述機能という）。デジタル回路の論理を検証するためのシミュレータにおいては、通常、ハードウエ

としては、VHDLの「REDA」、「WRITE」等を使用することができ、上記(2)としては、VHDLにおけるジェネリックパラメタによる信号名の受渡機能を用いることができ、さらに、上記(3)としては、VHDLの「ATTRIBUTE」(属性定義コマンド)を使用することができる。以下の説明においては、ハードウェア記述言語としてVHDL用いた場合について説明する。

【0019】10、20は本発明において付加される部分であり、10はテストベクタ・パラメタ定義ファイル11と入出力機能付きI/Oセル・ライブラリ12からなる入力ファイル、20は非処理系ログファイル21および変換手段22からなる出力部である。入力ファイル部10のパラメタ定義ファイル11により、テストベクタにおける端子名とテストパターンとの対応関係等が定義される。また、入出力機能付きI/Oセルライブラリ12は、ICのI/Oセル(後述する)の機能をVHDL言語で記述したファイルであり、ここにパラメタを記述し、後述するように該I/Oセルライブラリ12をコンパイルした結果と、回路データのソースプログラム(設計データ)をコンパイルした結果を用いて論理シミュレーションを行い、非処理系ログファイル21に信号名等を反映させる。

【0020】また、出力部20の非処理系ログファイル21には、処理系52、53による論理検証結果の内、ICのI/Oセルの信号値がその信号名を付して格納され、これに基づき各社形式のテストベクタが作成される。上記I/Oセルの信号値は、ICの各ピンの信号値であり、双方向端子については、双方向I/Oセルを制御する制御信号の信号値も格納される。22は上記非処理系ログファイル21に格納されるI/Oセルの信号名、論理等を各社形式のテストベクタに変換する変換手段であり、変換手段22に、テストベクタヘッダ部分を生成するパラメタ31を与えることにより、変換手段22はICベンダ形式に応じたテストベクタを生成する。上記パラメタ31の記述は、前記したVHDL言語のテストベクタパラメタ記述機能を利用する。これにより、IC設計者は、CADツールや各社のテストベクタ形式を意識することなく、テストベクタヘッダ部分に対応したパラメタを記述することができる。

【0021】なお、各処理系52、53に設けられたシミュレータログファイル52d、53dには、シミュレーション結果の全ての信号値(入出力信号だけでなくICの内部素子の各信号値も含む)を落とすことが可能であるが、テストベクタ作成に必要なデータはICの入出力ピンの信号値でありシミュレーション結果が全て必要なわけではなく、また、上記シミュレータログファイルの形式はCADツール毎に異なっているため、シミュレータログファイルから各社形式のテストベクタを作成するには、前記図7に示したように各CADツール毎の変

【0022】これに対し、本発明のように、非処理系ログファイル21を設け、ここに、ICのI/Oセルの信号名、信号値等を格納することにより、テストベクタ作成に必要なデータを格納したシミュレータの形式に依存しないログファイル22を生成することができる。そして、変換手段22に、パラメタ31の指定に応じて上記ログデータを各社形式のテストベクタに変換する機能を持たせておけば、上記パラメタ31を与えるだけで各社形式のテストベクタを作成することができる。しかも、入出力機能付きI/Oセルライブラリ12の記述、回路データの記述、上記パラメタ31の記述等をVHDL言語で統一的行なえるようにすれば、IC設計者は、処理系52、53の違い、各社ベクタ形式の違い等を意識することなく各社形式のテストベクタを作成することができる。

【0023】図3はLSIのI/Oセルモデルの一例を示す図である。図3に示すように、LSIは、通常、所望の論理演算等を行う論理機能回路と、外部ピンから与えられる入力信号を上記論理機能回路に与える入力セル、論理機能回路の出力を外部ピンへ出力するための出力セル、および、制御信号により信号方向が制御される双方向セル等を含むI/Oセルを備えている。テストベクタ作成に必要なデータは、上記I/Oセルの信号名(例えば同図におけるi1、o1)とその信号値である。また、双方向セルの場合には、その信号値とともに信号方向(入力状態か出力状態か)が必要となる。双方向セルの信号方向は、双方向セルに与えられる制御信号(同図のc1)の信号値により定まるので、双方向セルについて必要なデータは、各信号の信号名(例えばb1、c1)とその信号値である。本実施例においては、図3に示すように、入出力機能付きI/Oセルライブラリ12に記述されたパラメタを前記したVHDLのパラメタ受渡機能で処理系52、53の出力に反映させ、前記したVHDLの入出力機能を利用して、処理系52、53の論理検証結果とともに信号名を非処理系ログファイル21に書き込む。

【0024】図4は図2に示した実施例のシステムの動作を説明する図であり、図3、図4により本実施例における変換処理手順を説明する。まず、I/Oセルライブラリ(ソースプログラム)31に設計レベルへ渡すためのパラメタを記述する。次に、該I/Oセルライブラリ31を用いて回路設計を行い、コンフィグレーション・ファイル(設計回路のソース)32を作成する(ステップS1)。これにより同図に示すように、パラメタに各I/Oセルの実際の信号名i1(入力セルの信号名)、o1(出力セルの信号名)、b1(双方向セルの信号名)、c1(双方向セルの制御信号の信号名)等が入ったコンフィグレーション・ファイル(ソースプログラム)32が作成される。次に上記コンフィグレーション

処理系によりコンパイルし、I/Oセルライブラリのオブジェクト・プログラム 33 と設計回路のオブジェクト・プログラム 34 を得る (ステップ S 2)。そして、上記オブジェクト・プログラム 33、34 を用いて既存の処理系により論理シミュレーションを行い (ステップ S 3)、前記した入出力機能を用いて論理検証により得た信号値を信号名とともに非処理系ログファイル 36 に書き込む。

【0025】非処理系ログファイル 36 に書き込まれるデータは、同図に示すように、各 I/Oセルの、各時点における信号値 (シミュレーション結果) と、信号名である。同図の例においては、各時点 (0, 50, 80, …) におけるクロック端子 CLK の信号値と、出力セル o1 の信号値と、入力セル i1 の信号値と、双方向セル b1 の信号値と、双方向セルの信号方向を示す制御信号 c1 が示されている。なお、同図では、双方向セルの制御信号 c1 が途中でハイレベルからローレベルに変化する場合が例示されており、この時点で双方向セル b1 の信号方向が変化している。上記のように非処理系ログファイル 36 を作成するとともに、VHDL 言語のテスト

【0026】

【発明の効果】以上説明したように、本発明においては、回路表現/ハードウェア記述言語が持つパラメタ受渡機能、入出力機能を用いて、設計回路の I/Oセルの信号値、信号名等を格納した非処理系ログファイルを生成し、非処理系に格納されたデータと、指定された IC ベンダ情報に基づき各種形式のテストベクタを作成するようにしたので、回路設計者は、使用する論理シミュレータ、各ベンダ形式を意識することなく、各種形式のテストベクタを作成することができる。このため、従来のように各ベンダ形式に応じた複数の変換手段を設け、使

用する論理シミュレータ、各ベンダ形式に応じて該複数の変換手段の中から必要とする変換手段を選択する必要がなくなり、また、一つの表現および手順で各種のテストベクタを作成することができるので、IC 設計者の負担を大幅に低減化することができる。

【図面の簡単な説明】

【図 1】本発明の概略構成を示す図である。

【図 2】図 2 は本発明の実施例のシステムの構成を示す図である。

【図 3】LSI の I/Oセルモデルの一例を示す図である。

【図 4】本発明の実施例のシステムの動作を説明する図である。

【図 5】IC ベンダが要請するテストベクタ形式の概略構成を示す図である。

【図 6】IC ベンダが要請する他のテストベクタ形式の概略構成を示す図である。

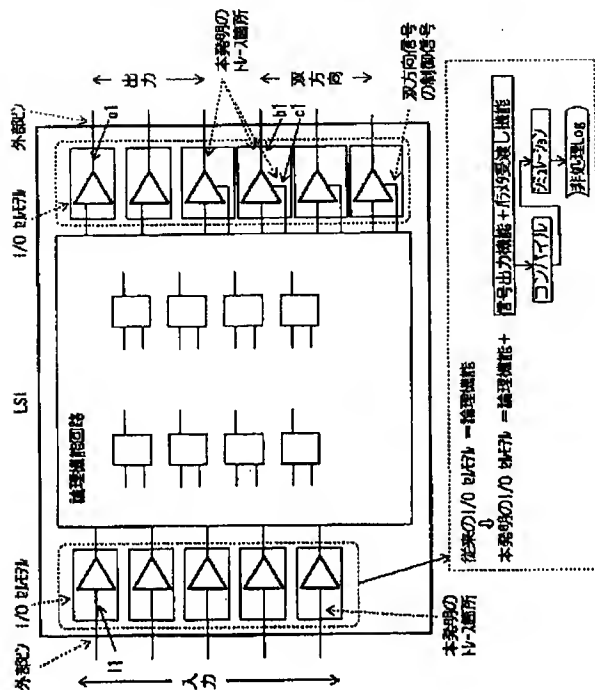
【図 7】従来の変換システムの構成を示す図である。

【符号の説明】

- | | |
|----------|--------------------|
| 1 | I/Oセルライブラリ |
| 2 | 回路データを格納したファイル |
| 3 | 論理シミュレータ |
| 4 | 非処理系ログファイル |
| 5 | テストベクタパラメタのソースファイル |
| 6 | 変換手段 |
| 51 | 回路データ・ファイル |
| 52, 53 | 第1および第2の処理系 |
| 52b, 53b | コンフィグレーションデータ |
| 52a, 53a | コンパイラ |
| 52c, 53c | シミュレータ |
| 52d, 53d | シミュレータログファイル |
| 10 | 入力ファイル |
| 20 | 出力部 |
| 11 | パラメタ定義ファイル |
| 12 | 入出力機能付き I/Oセルライブラリ |
| 21 | 非処理系ログファイル |
| 22 | 変換手段 |

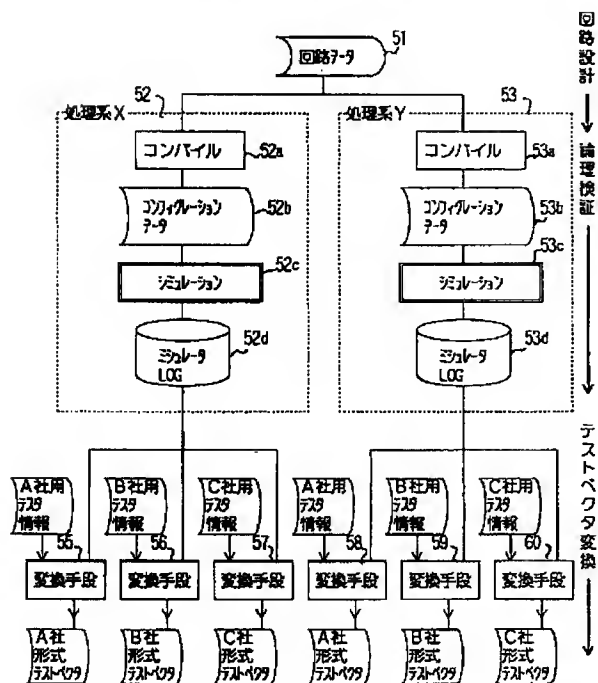
【図3】

LSIのI/Oセルモデルの一例を示す図



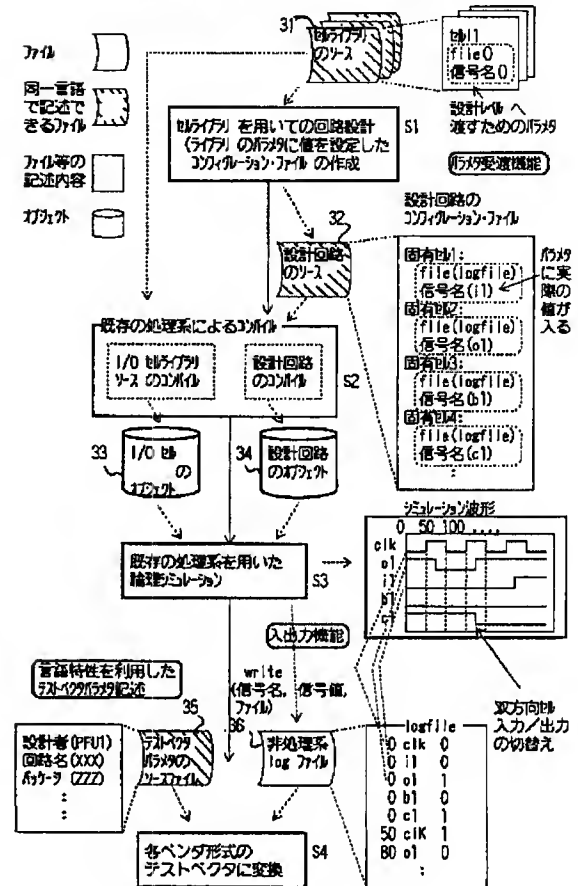
【図7】

従来の変換システムの構成を示す図



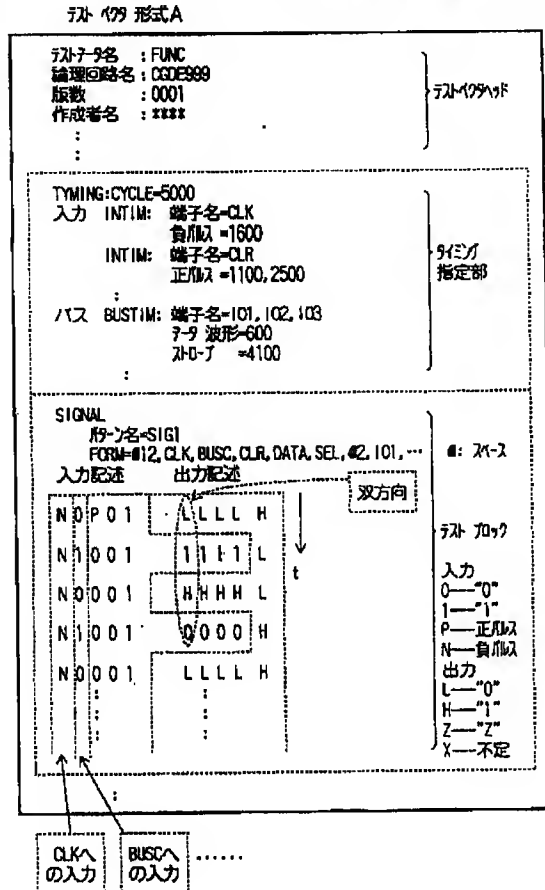
【図4】

本発明の実施例のシステムの動作を説明する図



【図5】

I Cベンダが要請するテストベクタ形式の概略構成を示す図



【図6】

I Cベンダが要請する他のテストベクタ形式の概略構成を示す図

